

One-layer particle level set method

Lanhao ZHAO^a, Hongvan KHUC^a, Jia MAO^a, Xunnan LIU^a, Eldad AVITAL^b

^a College of Water Conservancy and Hydropower, Hohai University, Xikang Rd., Nanjing 210098, PR China

^b School of Engineering & Materials Science, Queen Mary University of London, Mile End Rd., London E1 4NS, UK

ABSTRACT

One-layer particle level set (OPLS) has been developed by using Lagrangian particles that are employed to correct both advection and re-initialization procedures of the level set function. In which, a level set function is utilized to smooth physical properties of the interface, while one-layer Lagrangian particles are used to track the interface directly. This method exhibits excellent mass conservation properties compared to the LS method. As a special aspect, the OPLS method enables management of merging and stretching of interface in an effective way. This capability is similar to the particle level set (PLS) method. However, the new approach of the OPLS method offers a more straightforward technique. This approach is validated with classical benchmark test cases, such as the long term advection of a circle, rotation of a slotted disk, single vertex in a box, merging and separating of circle. The results from the proposed method show good agreement with the numerical experiments published results and the OPLS method is verified to be highly reliable and accurate.

KEY WORD: *Lagrangian particle; Level set method; Re-initialization; Particle level set method; Reseeding schemes.*

* Corresponding author:

JIA MAO

Tel.: +86 15150654242

Fax: +86 25 8373 1332

E-mail address: maojia@hhu.edu.cn

1. Introduction

Numerical simulation of an interface between different fluids plays a crucial role in a variety of fields, including engineering and other types of scientific area. Difficulties of these simulations arise from the arbitrary moving interface, particularly for those that undergo extreme topological changes, e.g., merging or breaking. Therefore, interface representation with accurate and robust numerical schemes has become a very active field of research.

Various methods have been proposed to improve the simulation of moving interfaces. These methods can be divided into the Lagrangian methods and the Eulerian methods, is based on the mathematical descriptions of fluid flow. The Lagrangian method frame uses particles transported with fluid to describe interfaces. This method can enhance the conservative property without advection errors within the entire calculation process. However, clear deficiencies have been pointed out by many existing researcher [1-6], particularly in complex free surface such as thin boundary layers, merging and stretching region. Particles cluster is caused by tensile instability and lack of consistency by the domain distortions. To address this problem of the Lagrangian method, incorporation of the Lagrangian method with the Eulerian method in the arbitrary Lagrangian-Eulerian (ALE) approach is another way to simulate interface deformation. This method is known to be very efficient in the calculation of flows involving curved or moving boundaries. However, in extreme deformations such as wave breaking, re-meshing is inevitable.

Eulerian methods attracted much attention, because it could address some of the dis-efficiencies of the Lagrangian methods. The existing Eulerian methods can be divided into interface tracking methods and interface capturing methods [7, 8]. The interface tracking methods solve the interface evolution by either setting marker points or adjusting the interface to attach the mesh. Belonging to interface tracking methods, The Marker-And-Cell (MAC) method proposed by

Harlow and Welch [9, 10] is the first attempt to treat flows with complex interfaces. MAC technique should be capable to model highly deformed fluids. The technique can also manage merged and detached fluid. However, the MAC method is computationally expensive since every marker-particle's position must be stored over the fluid domain. In addition, oscillations on the interface may occur due to markers that move independently when the interface is reconstructed [11-13].

The other kind of Eulerian methods namely interface capturing methods have gained more popularity as it could easily calculate characteristics of the interface in a merging or stretching region [14]. In interface capturing methods, a function is usually employed to describe the advection of the interface implicitly. Among the proposed interface capturing methods, the Volume of Fluid (VOF) method and the Level Set (LS) method have drawn the most attention in recent years. The VOF method describes interface by color function. Even though the VOF method displays accurately mass conservation [15, 16], this method still encounters difficulties in an accurate presentation of the interface's curvature because of smearing in the color function. To overcome this problem, Osher and Sethian [17] proposed the LS method. In the LS method, the smooth nature of the signed distance function can help to overcome several drawbacks of the other interface describing methods [17-21]. Additionally, the LS method can automatically handle merging and breaking of interfaces. Unfortunately, due to convection and re-initialization of LS function, loss/gain of mass will accumulate in the calculation process [21-23].

Considering the simplicity and the accuracy of interface representation, many researcher have concentrated on a re-initialization procedure of level set equation [24-26] while the sign distance function was adjusted during the simulation. Elin and Olsson [27, 28] improved the LS method itself to the Conservative of Level Set (CLS) method. The main idea of CLS method is replacing

the sign distance function of the standard method with a hyperbolic tangent profile, which is advected in a conservative way. Compared with the standard LS method, the CLS method exhibits drastically improved mass conservation properties and has been successfully employed in many applications [29-35]. However, the re-initialization procedure never completely satisfy the mass conservation [36]. The results can be seen in the first two numerical experiments of this paper.

Hybrid schemes of the LS method to show advantage with other conservative front capturing methods have been suggested. One of the hybrid approaches is the improved conservative level set (ICLS) method, which combines the LS method with the CLS method. This method has a good mass conservation property and gives accurate interface information [21]. Another common method was introduced by Sussman and Puckett [37], which combines the LS and VOF (VOF/LS) method. In this approach, the interface is reconstructed from volume fractions to ensure mass conservation while normal vectors from the LS function are used for better approximation of geometrical quantities [26-28]. However, the drawback of both the CLS method and the VOF method combined with the LS method is reduced accuracy in the representation of the interface as compared with pure Lagrangian method [22, 23].

Recently, the Lagrangian approach is used to modify the LS, and that approach has gained favor. One of the successful refined algorithms is the particle level set (PLS) method which was introduced by Enright [38]. The PLS method combines the accuracy benefits of the Lagrangian front tracking with the simplicity and efficiency of the LS method. This method represents interface characteristics more accurately in comparison with the pure LS method. Conservation improvement lies in the correction of interfaces by escaped Lagrangian particles. Even though there are several wide applications [39-48], the PLS method still has complicated reseeded of

particles, the method is also based on various artificial tuning. Thus, the final results can vary to the sensitivity of different selections. Additionally, an escaped particle normally can only correct the level set value on one side of the interface, while it hardly affects the value on the other side [39, 49]. Unfortunately, the normal vector is calculated by the uncorrected level set function which could result in an inaccurate representation of the interface, especially in an under-resolved region [46, 49, 50].

In order to avoid the drawbacks of the PLS method mentioned above, a novel method, One-layer Particle Level Set (OPLS) method is proposed in this paper. Compared with the PLS method, the number of particles is reduced remarkably in the OPLS method since the Lagrangian particles describe accurate interface through only one-layer particles (Figure 1). The New approach of the OPLS method offers a straightforward technique for reseeding process of particle even in merging or stretching region of interfaces with high stability. The motivation is based on the fact that Lagrangian particles are used to track the interface position directly, while the LS method is applied to represent the interface smoothly. This method is a novel improvement of the LS method and it is totally different from the previous PLS method and others.

This paper is organized as follows: basic idea and the motivation of OPLS method are given in Section 2, numerical approach of OPLS method has been discussed in Section 3. In Section 3, this new method has been introduced in detail including the modification of the interface and the presentation of merging/stretching interfaces by deleting/adding particles. In order to show the accuracy and robustness of the present method, the OPLS method is applied to calculate two benchmark test cases; three new numerical experiments and the current results are compared with those in published literature in Section 4. Finally, concluding remarks are summarized in Section 5.

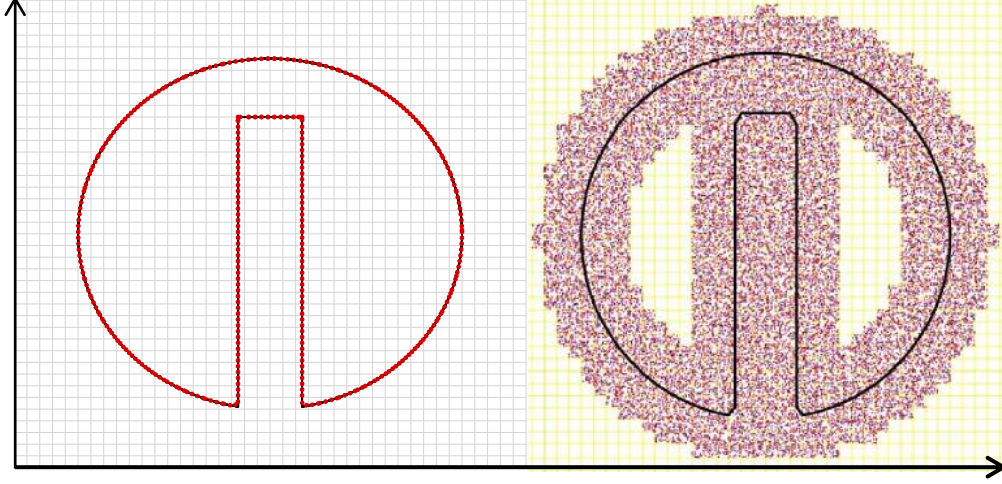


Figure 1. Left: one-layer particles placement (red) and actual interface (black) in OPLS method. Right: multi-layer particles location and actual interface (black) in PLS method (Enright [38]).

2. Motivation

2.1 Basic idea of particle level set method

Particles generation

The PLS method combines the best properties of Eulerian LS method with a marker particle Lagrangian scheme. A set of marker particles are placed both side near the interface. Positive particles are placed in the region which satisfies Ω^+ , and negative particles are located in the Ω^- region. The particles are used to correct the error of interface and to remove errors in the level set function in under-resolved region. A given number of particles of each sign (i.e., 4 in 1D, 16 in 2D and 64 in 3D) are generated not only in each interface cells but also both sides near the interface cells. Then particles are attracted to the correct side of the interface with positive particles to the Ω^+ region and negative particles to the Ω^- region.

Particles correct level set function

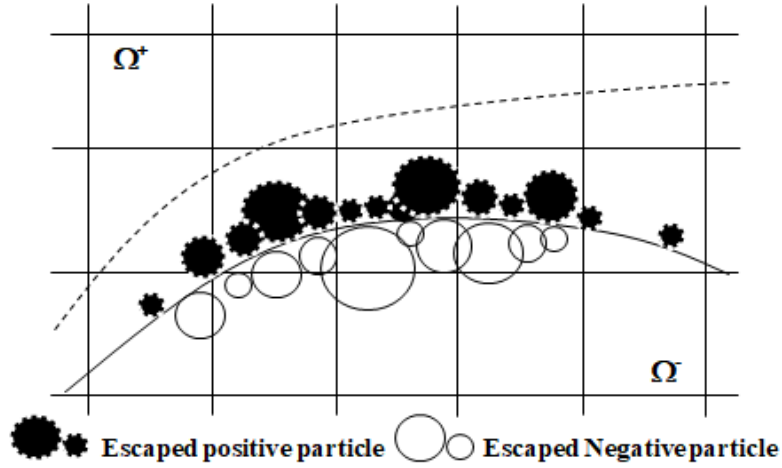


Figure 2. Dash line is the level set function before modification. Particles are escaped in both side to rebuild the level set function (solid line) by the PLS method.

Escaped particles are used to reconstruct the level set function in under-resolved regions. A particle is forced as escaped when it crosses the interface by more than its radius. Figure 2 illustrates the idea of an escaped particle to correct the level set function of PLS method. The escaped positive particles are used to rebuild the Ω^+ region. In Ω^- region, the process is carried out with escaped negative particles.

Particles reseeding

Particle reseeding is a process in which particles are dynamically added and deleted in the region with highly deformed interfaces. Reseeding is necessary to maintain resolution of the interface during merging or stretching time. The idea of particles addition and deletion has been addressed by many authors [38, 46, 49-52]. Reseeding is carried not only by adding and deleting particles in cells near the interface, but also deleting particles which have drift far from zero level set due to inaccurate information of the interfaces. Additionally, during the reseeding process, operation should be defined by the acceptable number of particles per grid cell by upper and lower bounds.

If either bound is exceeded, particles addition and deletion may occur to keep its initial value. In the original method that was introduced by Enright in [38], reseeding process is carried out by:

Step 1: Identification of all the no-escaped particles in each cell.

Step 2: Utilization of local value of the level set function is used to decide:

- Delete all the non-escaped particles (if a cell is not near the interface).
- Otherwise, particles are added to the cell then attracted to the interface, in case a cell is near the interface (within three grid cells) and currently has less particles than the previously defined maximum (i.e., 4 in 1D, 16 in 2D and 64 in 3D).

2.2 Motivation of one-layer particles level method

Even though the PLS method applies Lagrangian particles to correct interfaces, it still has two main drawbacks: long time calculation caused by a large number of particles and the complicated reseeding of particles. Many improvements that focused on particle modification have been suggested and these researcher also improved the randomization and artificialness of the particles modification [23, 45, 47].

To overcome the drawbacks of PLS method, a new approach, OPLS method, is proposed in this paper. In the OPLS method, Lagrangian particles can determine the precise interface position without allocating abundant particles or apply complex modification methods. Only one-layer particles are generated on the interface then, at each time step, these particles can move to accurate positions for the purpose of representing the interface. In order to distinguish different phases and to calculate the physical properties near the interface, traditional LS method is employed. As mentioned above, LS method has difficulties in mass conservation during the advection procedure and in the re-initialization procedure. Even though the inaccuracy due to the

advection process can be avoided by using high-order discretization schemes, this kind of inaccuracy could induce mass loss/gain and will accumulate during the process.

The motivation of the OPLS method is improvement to the LS method since the Lagrangian particles are used to describe exactly the interface and correct both the advection procedure and the re-initialization procedure of the level set function directly. In the OPLS method, LS method is applied to represent the interface smoothly, and Lagrangian particles are used to modify the interface directly. This approach is similar to the PLS method when simulating interfaces that require special handling, such as, adding particle when the interface undergoes stretching. Besides, when interfaces are merged, deleting particles are needed. However, the OPLS method allows simulation of interface by a more straightforward technique. This principle has been described in the next section in detail, specific practices and procedures of OPLS method are summarized below:

Step 1: *Generating particles:*

Particles are initially scattered on both sides of the interface, but do not distinguish between positive and negative. Particles are then attracted to the interface.

Step 2: *Updating position of particles:*

- Particle speed is interpolated linearly from the surrounding velocity field of the grid nodes.
- The position of the particle is updated by Third-order (Runge-Kutta) schemes to get accurate particle position.

Step 3: *Solving both advection and re-initialization procedures of level set function*

Step 4: *Correcting interface*

Particles are used to the interface modification. At each time step, both advection and re-initialization procedures of the level set function are modified.

Step 5: *Adding/deleting particles*

In accordance to the updated interface, the particle addition or deletion has been determined.

Then specifically, modification procedures of the OPLS method to correct moving interface has been explained in detail in Section 3.

3. Numerical aspects***3.1. Particle generation***

It should be noted that not all cells near interface should contain particles. Initially, the interface cells are detected. A cell is defined as an interface cell if the level set function value ϕ of its four nodes is not within the same sign. When the interface cells are found, particles are initially scattered on both sides of the interface cells but do not distinguish between positive and negative. Along normalized coordinates of grid nodes ξ, η in each cells, the number of particle N is given. Such as, $N=1$, 1 particle is generated in the center of cell; $N=2$, 2 particles are generated each direction and $N=3$, each direction contains 3 particles (see Figure 3).

In the OPLS method, each particle stores its position is advected with the flow using the Lagrangian particle Eq. (1) which is also used in PLS method [38]:

$$\frac{dx_p}{dt} = u(x_p) \quad (1)$$

where x_p is the position of the particle P and $u(x_p)$ is its velocity. Particle speed is interpolated linearly through the surrounding velocities field of the grid nodes. Third-order accurate TVD-Runge-Kutta schemes are used to update position of the particle over time.

In order to attract a particle P to the right interface, i.e., from a current interpolation level set function value φ_p is attracted to φ_{goal} along the shortest possible way, the attraction equation is shown in Eq. (2):

$$x_{new} = x_p + \lambda(\varphi_{goal} - \varphi_p)\vec{n}(x_p) \quad (2)$$

where the parameter λ is set to 1 at the beginning and is successively halved until Eq. (2) places the particle within the expected domain. The normal vector \vec{n} gives the direction of the nearest particle to the interface. To obtain a random distribution of the particles along the direction normal to the interface, each particle is assigned a goal value of $\varphi_{goal} = 0$. If the particle does not lie in the desired interface region after several inner iterations, the particle must be deleted.

Figure 3 describes how the particle are generated by the given particle number per cell and then, they are attracted to the right interface at initial step. The number of particles placed in each cell could be adjusted according to the desired interface resolution. The suggestion particles of each set per cell $N = 2$ should be enough for an accurate interpretation of the moving interface in two phase flow.

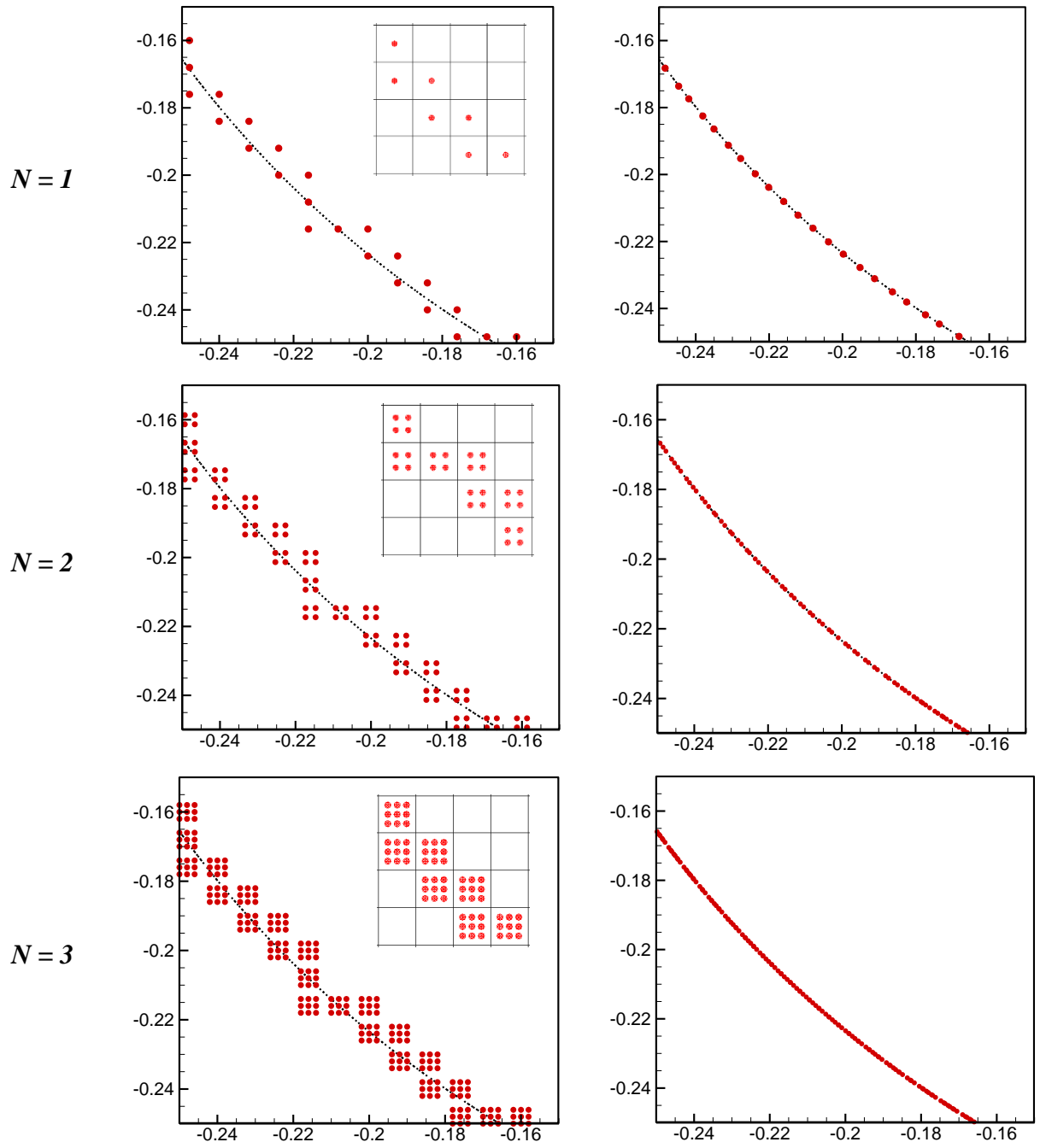


Figure 3. Particles are generated then are attracted to the interface at the initial step in mesh size 100 . Black dash line represents the interface.

3.2. Modification of level set function

The level set function φ in the OPLS method is defined as:

$$\varphi(x) > 0, x \in \Omega^+; \varphi(x) < 0, x \in \Omega^-; \varphi(x) = 0, x \in \Gamma \quad (3)$$

where Ω^+ and Ω^- are a positive and negative region respectively separated by interface Γ in computational Ω domain. The evolution of the level set function φ is given by the standard LS equation:

$$\frac{\partial \varphi}{\partial t} + \mathbf{u} \cdot \nabla \varphi = 0 \quad (4)$$

The re-initialization equation is as follow:

$$\frac{\partial \varphi}{\partial \tau} + \text{Sign}(\varphi^n) |\nabla \varphi| = \text{Sign}(\varphi^n) \quad (5)$$

where τ is fictitious time for the convenience of calculation and Sign is the sign function.

Both particles and level set function are advected by the local velocity field. During the advection and re-initialization procedure of the level set function according to Eq. (4) and Eq. (5) particles positions Eq. (1) are exactly located on the interface. The problem is an error in the level set function due to a drift in the value of φ . This phenomenon illustrates that the LS method can lose the accuracy during the movement. In order to counteract this problem, the level set function value is modified by just one-layer particles in the OPLS method.

While an interface is moving, the particles positions are transported over time and the particles lie in a new interface. The level set function value of these particles is interpolated from the surrounding the level set function value of the grid nodes as:

$$\varphi_p^k = \sum_{i=1}^n N_i^k \varphi_i^k \quad (6)$$

where N_i^k is the shape function [53], is calculated by Eq. (7) with ξ, η are the normalized coordinates and ξ_i, η_i are the normalized coordinates at the grid nodes. φ_i^k is the level set function value of the surrounding the grid nodes i at time $k, i = (1, \dots, n)$, n is the number of grid nodes.

$$N_i = \frac{1}{4}(1 + \xi_i \xi)(1 + \eta_i \eta) \quad (7)$$

Particles follow the velocity field and location of each particle is correct. But, an error in the level set function value φ can occur when performing the re-initialization due to level set function value of particle $\varphi_p^k \neq 0$. Considering the accuracy of the interfaces represented by the Lagrangian particles, it is necessary to modify the level set function value φ_i^k to ensure $\varphi_p^k = 0$. Therefore, a certain number of iterations is needed in the process of the re-initialization for time k . The modification process of OPLS method is described below with a help of Figure 4.

First, the level set function value of each particle is interpolated by the surrounding level set function value of the grid nodes:

$$\varphi_p^{k,j} = \sum_{i=1}^n N_i^k \varphi_i^{k,j} \quad (8)$$

where j stands for a certain number of iterations at time k , $\varphi_i^{k,0}$ is the level set function value of the surrounding grid nodes i of the in the initial stage.

Then, if any of the level set function value of particle does not set to zero, a next loop $j+1$ is required to distribute the error $\varphi_p^{k,j}$ by the surrounding level set function value of the grid nodes following these equation:

$$\varphi_i^{k,j+1} = \varphi_i^{k,j} - N_i^k \varphi_p^{k,j} \quad (9)$$

$$\varphi_p^{k,j+1} = \sum_{i=1}^n N_i^k \varphi_i^{k,j+1} \quad (10)$$

As well as, the modification process at time k in one interface cell of Figure 4 illustrates that, although the level set function value of particle P_1 ($\varphi_{P_1}^{k,j} = 0$) after finish loop j but others do not offset to zero. Thus, all these particles must be put into the next loop $j+1$ of Eq. (9) and Eq. (10) to distribute the errors of level set function value $\varphi_p^{k,j}$. It should be noted that obtaining $\varphi_p^{k,j} = 0$ is a process of linear interpolation. After Eq. (10) was controlled, the level set function value $\varphi_p^{k,j+1}$ of all the particles may not offset to zero. Thus, repetition of the iteration is required until the norm of level set function value of particle $\varphi_p^{k,j}$ is smaller than a given tolerance value ε ($\|\varphi_p^{k,j}\| \leq \varepsilon$) or a certain number iteration also is requested in complex problems. In numerical experiments of Section 4, the tolerance value $10^{-3} \leq \varepsilon \leq 10^{-2}$ depending on the grid mesh and at almost test use two iterations while circle separation test needs ten iterations.

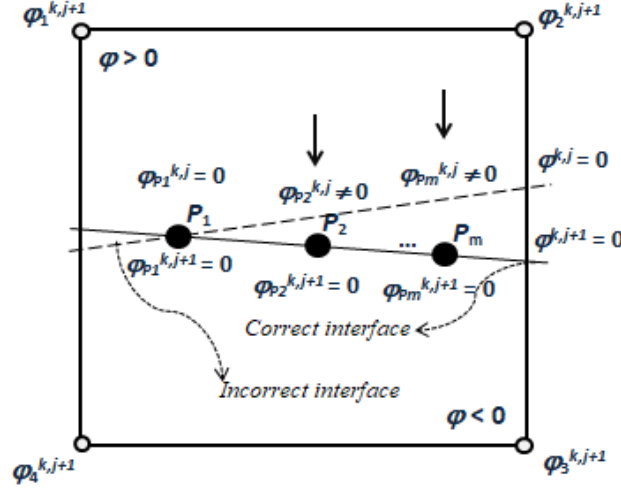


Figure 4. Because of the performing the re-initialization due to level set function value of particle drift to incorrect interface. OPLS method allow particles (black bold) modification error of level set function value by a certain number of linear interpolation surrounding level set function value of grid nodes.

With this new re-initialization procedure, the disturbances due to the particle correction can be solved and more accurate interface representation can be simply obtained. Compared with “escaped particle” proposed by Enright [38] or others re-initialization procedure in [49, 50], the advantage of this technique in the OPLS method is a more straightforward technique that automatically and remarkably reduces the number of particles affecting the computation cost.

3.3. Adding particles

In some situations (for example, in a high stretching region) an inadequate number of particles could make the re-initialization erroneous due to numerical oscillations or other drastic changes. The new interfaces are generated and particles should be added in each interface cells. The OPLS method can be completed by the particles added strategically with the following three steps:

Step 1: Define the interface cells in each time step if the level set function value φ of the four nodes is not within the same sign or cells have less particle, i.e., at least 1 particle per cell, these cells are defined as target cells.

Step 2: Generate particles in each interface cells.

Step 3: Attract particles to the interface.

As can be seen in Figure 5, when an interface from the dash line is stretching to the solid line, new interface cells are found (blue dark cells). By a given number of particle per cell N , the OPLS method allows automatic generation of new particles in these cells. Unfortunately, these new particles are not directly on the right interface. Therefore, the attraction function in the OPLS method puts them to correct place. In comparison, the attraction step in the OPLS method has some similarities with the PLS which was introduced by Enright [38]. However, the attraction step in PLS is distributed for holding particles in a narrow band near the interface while the OPLS method uses this function for only particle located on the interface cells.

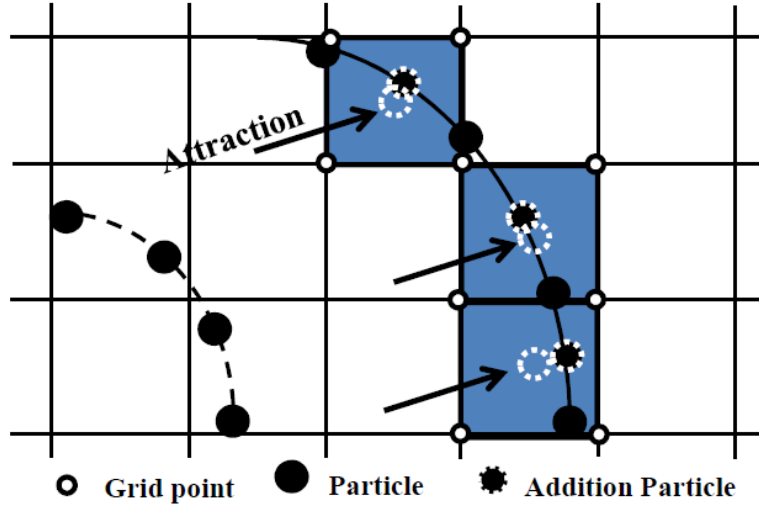


Figure 5. The strategy of adding particles; Dashed line: interface ($\varphi = 0$); Solid line: interface after stretching. Blue cell: interface cell contains less particle; Black circle: Particles; White circle: nodes grid; Black and white around: Added particle; White dash line: Particles are generated in the interface cells.

3.4. Deleting particles

In each step, particles have just been advected by the velocity field. When interfaces merge, the initial interface disappears and some particles are left alone. Therefore, deletions of these

particles are needed. With the OPLS method, particles deleting strategy has been taken into consideration. The strategy is summarized in two steps listed below:

Step 1: Generate imaginary points P_1 along the normal vector $\vec{n} = \frac{\partial \varphi}{|\partial \varphi|}$ with a certain delta distance and point P_2 on the reverse direction.

Step 2: Delete particle P , if values of $\varphi(P_1)$ and $\varphi(P_2)$ are of the same sign.

The particles deletion algorithm implemented in the OPLS method is based on the level set function value φ of each particle. Figure 6 illustrates the method of particle deletion in the merging region. In Figure 6.a) $\varphi(P_1)$ is positive while $\varphi(P_2)$ is negative. It means particle P still lies on the interface. Hence, particle P will be kept. However, in Figure 6.b) both values of $\varphi(P_1)$ and $\varphi(P_2)$ are negative. Which means an initial interface disappears after two interfaces are merged, thus particle P is left alone and does not represent an interface at this time. Therefore, particle P must be deleted.

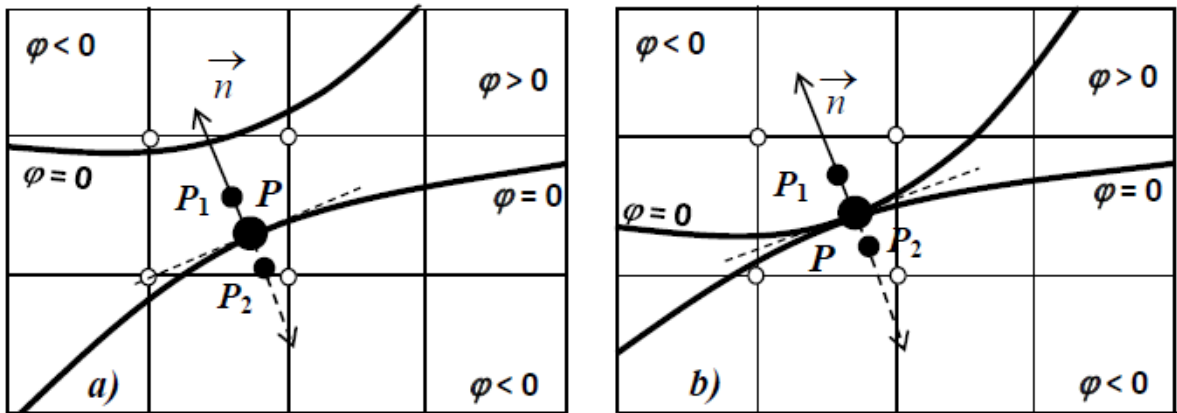


Figure 6. The strategy of particles deletion; Dashed line: interface ($\varphi = 0$); Big black circle: Particle point; Small black circle: Imaginary particles point; White circle: nodes of interface cell.

3.5. Solution of OPLS equation

Temporal discretization

A third order TVD-Runge-Kutta scheme [54, 55] is used to evolve the particle positions forward in time as follows, where L stands for the differential operator.

$$\begin{cases} \varphi^{(1)} = \varphi^{(0)} + \Delta t L(\varphi^{(0)}) \\ \varphi^{(2)} = \frac{3}{4}\varphi^{(0)} + \frac{1}{4}(\varphi^{(1)} + \Delta t L(\varphi^{(1)})) \\ \varphi^{(3)} = \frac{1}{3}\varphi^{(0)} + \frac{2}{3}(\varphi^{(2)} + \Delta t L(\varphi^{(2)})) \end{cases} \quad (11)$$

Spatial discretization and compact form of the OPLS method

In order to calculate the spatial discretization, the moment and the level set function are advected using the second order of two-step Taylor-Galerkin algorithm [56, 57]. Thus the final form of the scheme in the OPLS method can be summarized in the following compact form:

For two incompressible fluids separated by the interface Γ in a domain Ω , Eq. (4) of the level set function is advected by the fluid velocity field can be written as:

$$\frac{\partial \varphi}{\partial t} + \nabla \cdot (\mathbf{u}\varphi) = 0 \quad (12)$$

A re-initialization equation Eq. (5) for a few steps infictitious times τ also can be written as:

$$\frac{\partial \varphi}{\partial \tau} + \nabla \cdot (\mathbf{u}\varphi) = \text{Sign}(\varphi^n) + \varphi \nabla \cdot \mathbf{u} \quad (13)$$

where $\mathbf{u} = S(\varphi^n) \frac{\nabla \varphi}{|\nabla \varphi|}$

Employing two notations, the convective flux \mathbf{F} and the source term S for the above two equations of OPLS method:

Eq. (12): $\mathbf{F} = \mathbf{u}\varphi$; $S = 0$

Eq. (13): $\mathbf{F} = \mathbf{u}\varphi$; $S = -\text{Sign}(\varphi^n) - \varphi(\nabla \cdot \mathbf{u})$

The conservative convection equation can be written as:

$$\frac{\partial \varphi_i}{\partial t} + \nabla \cdot \mathbf{F} + S = 0 \quad (14)$$

where φ_i is an unknown level set value of grid nodes i .

A second order Taylor expansion of $\varphi_i^{k,j+1}$ is given in the neighborhood of $\varphi_i^{k,j}$ at time k , j stands for the inner iteration.

$$\varphi_i^{k,j+1} = \varphi_i^{k,j} + \Delta t \left. \frac{\partial \varphi_i}{\partial t} \right|^{k,j} + \frac{1}{2} \Delta t^2 \left. \frac{\partial^2 \varphi_i}{\partial t^2} \right|^{k,j} \quad (15)$$

The first-order time derivative of $\varphi_i^{k,j}$ can be calculated by

$$\left. \frac{\partial \varphi_i}{\partial t} \right|^{k,j} = -(\nabla \cdot \mathbf{F} + S)^{k,i} \quad (16)$$

And the second-order time derivative of $\varphi_i^{k,j}$ is calculated as:

$$\left. \frac{\partial^2 \varphi_i}{\partial t^2} \right|^{k,j} = -\frac{\partial}{\partial t} (\nabla \cdot \mathbf{F} + S)^{k,j} \quad (17)$$

In order to consider an intermediate step between step k^j and k^{j+1} , the OPLS method uses a two-step predictor-corrector Taylor-Galerkin procedure. The first step is a *prediction step* to calculate the solution at a time $k^{j+1/2}$ and the second one is a *corrector step* to carry out the solution of k^{j+1} . Hence, Eq. (14) can be simulated as:

Step 1: Finding a predicted value of $\varphi_i^{k,j+1/2}$ using the convective flux \mathbf{F} and source term S .

$$\varphi_i^{k,j+1/2} = \varphi_i^{k,j} - \frac{\Delta t}{2} (\nabla \cdot \mathbf{F} + S)^{k,j} \quad (18)$$

An approximate expansion of the convective flux $\mathbf{F}^{j+1/2}$ and source terms $S^{j+1/2}$ can also be written as:

$$\mathbf{F}^{j+1/2} = \mathbf{F}^j + \left. \frac{\partial \mathbf{F}}{\partial t} \right|^j \frac{\Delta t}{2}, \quad S^{j+1/2} = S^j + \left. \frac{\partial S}{\partial t} \right|^j \frac{\Delta t}{2} \quad (19)$$

which gives the first order time derivatives of the flux and sources terms:

$$\left. \frac{\partial \mathbf{F}}{\partial t} \right|^j = \frac{2}{\Delta t} (\mathbf{F}^{j+1/2} - \mathbf{F}^j), \quad \left. \frac{\partial S}{\partial t} \right|^j = \frac{2}{\Delta t} (S^{j+1/2} - S^j) \quad (20)$$

Step 2: Substituting the above equations into the second-order time derivative of Eq. (17):

$$\left. \frac{\partial^2 \varphi_i}{\partial t^2} \right|^{k,j} = -\frac{2}{\Delta t} [\nabla \cdot (\mathbf{F}^{j+1/2} - \mathbf{F}^j) + S^{j+1/2} - S^j] \quad (21)$$

The first time derivative $\left. \frac{\partial \varphi_i}{\partial t} \right|^{k,j}$ (Eq. (16)) and second time derivative $\left. \frac{\partial^2 \varphi_i}{\partial t^2} \right|^{k,j}$ (Eq. (17)) must be

incorporated into the Taylor expansion of Eq. (15) in order to get the value of $\varphi_i^{k,j+1}$.

The advantages of two-step Taylor-Galerkin algorithm are its relative simplicity and its ability to save computation cost. Second order accuracy for both in space and time is provided for structured or unstructured grids.

4. Numerical experiments

In the following examples, the percentage of relative global mass conservation error $E_r(t)\%$ in this paper is defined as:

$$E_r(t)\% = (1 - \frac{A_t}{A_0})\% \quad (22)$$

where A_0 is the initial value of the area at $t = 0$; A_t is the value of area enclosed at time t_i in the zero contour of level set function.

4.1. A long term advection of circle

A circle with a diameter of 0.15 is transported by a uniform flow of normalized velocity in a rectangular domain $[1 \times 1]^2$. The temporal evolutions by different methods are obtained in Figure 7. It can be seen in this figure that the interfaces calculated from the LS, CLS and ICLS method loss their original shapes and become asymmetric with rough surface around the notch by time 200. The rough profile near the notch is due to the coarse grid resolution and can be reduced by increasing the grid resolution. Among all the four methods, ICLS is fairly good, but OPLS method shows the best capability to keep the original shape and a fully mass conservation. Based on the above discussion, the performance of the LS method is very poor in terms of mass conservation, the reason for the mass loss in this case is due to the lack of re-initialization procedure. Shape preservation is highly dependent on the grid resolution. CLS method has good

mass conservation and shape preservation for the fine grid resolution but behaves badly when the grid becomes coarse. Thus, the accuracy of CLS and ICLS method also greatly depends on the mesh size and would require additional computational resources to handle complex flow problems. Overall, the OPLS method exhibits a better mass conservative feature and the interface profile matches with the exact solution perfectly for every grid resolutions. Results are compared with other four methods, shown in Figure 8. In order to verify the ability of OPLS in coarse meshes, set of resolution: (50×50) , (100×100) and (200×200) are compared to each other. As can be seen in Figure 9, the grid of (200×200) gets an excellent result, even coarse mesh grid (50×50) , the mass loss $E_r(t)\%$ is only up to 0.07%.

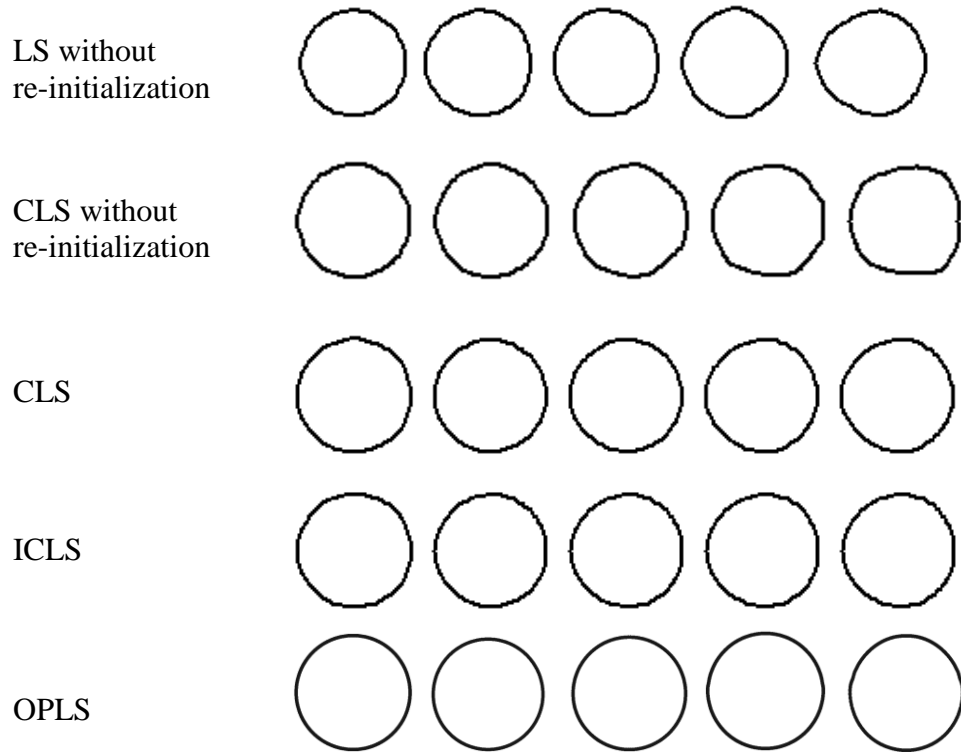


Figure 7. Long term advection of a circle under uniform velocity field; From top to bottom: LS, CLS, ICLS, OPLS; Time from left to right: $t = (0, 10, 40, 80, 200)$.

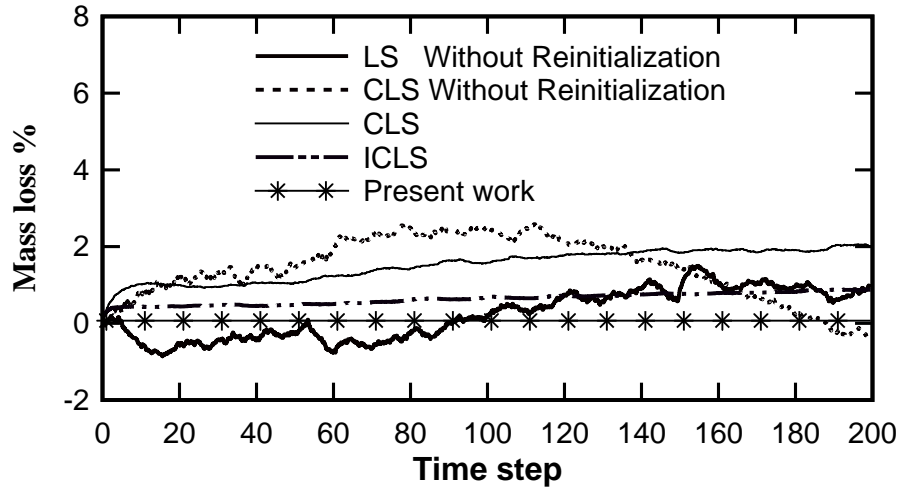


Figure 8. Mass loss in present work and other of Zhao [21].

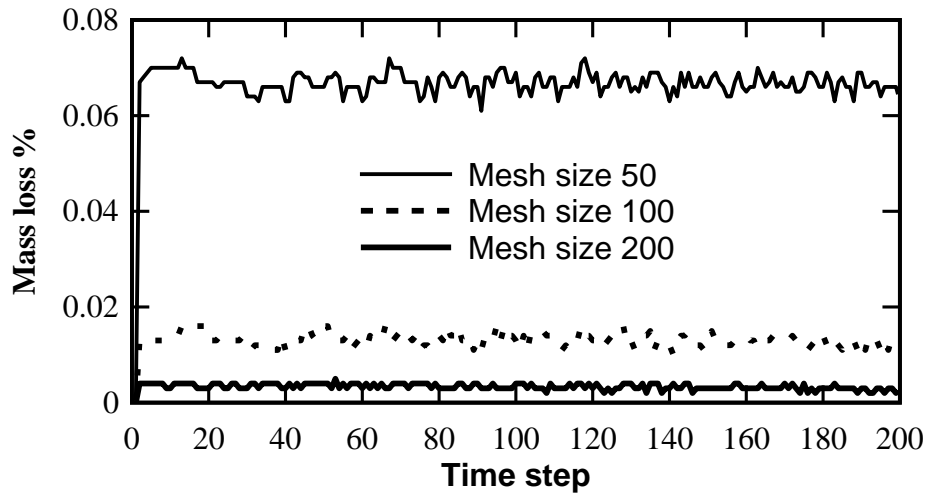


Figure 9. Mass loss of OPLS method.

4.2. Zalesak's disk

Zalesak's disk problem simulates a rotating slotted disk [58] which is one of the best known benchmark cases for testing the advection schemes. This classic problem is used extensively to test the advection ability of interface capturing methods and to characterize how accurately sharp

corners are transported. Zalesak's notched disk is defined as follows: Domain $[1 \times 1]^2$; Radius 0.15; Initial center $I(x,y)=(0.5, 0.75)$; Slot width 0.05; Slot depth 0.0125. The slotted disk is rotated under a velocity constant field is given as:

$$u(x,y)=-2\pi(y-0.5) ; v(x,y)=2\pi(x-0.5) \quad (23)$$

Eq. (23) represents a rigid body rotation with respect to (50×50) and the disk completes one revolution at one unit of time. In the simulation, sets of grid A(50×50), B(100×100) and C(200×200) were considered, the disk was allowed to complete two rotations and the results of different resolutions were gathered and compared with each other. Figure 10 illustrates the initial position of the interface represented by $\varphi=0$ (solid line) and location of particles before attraction step. After attraction, viewing sharpness of corners shows that particles are located exactly on the interface by only one-layer particles.

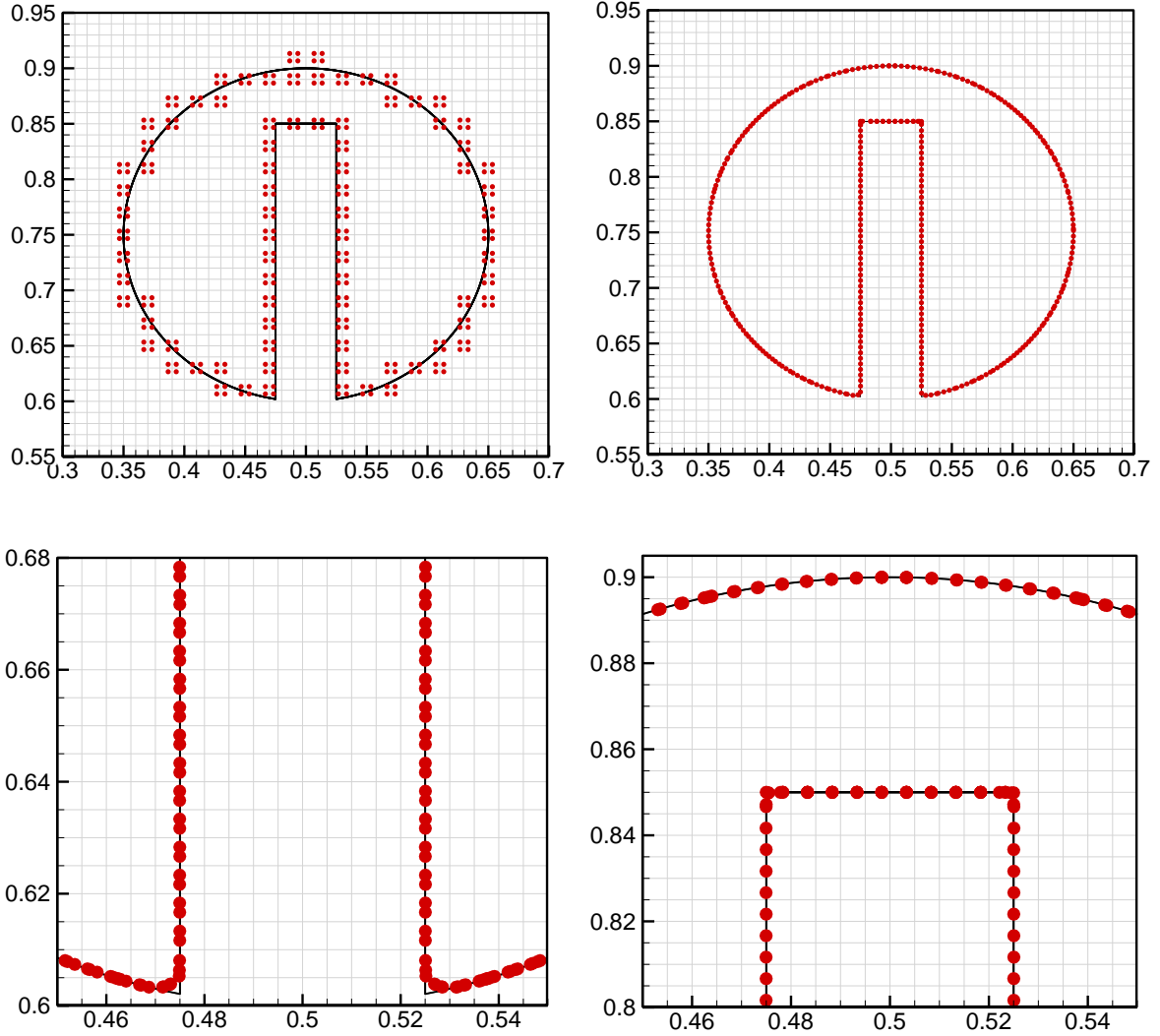


Figure 10. Top left: Particles (red) are generated around the interface (black solid line); Top right: One-layer particles are tracked into the interface; Bottom: Zoom at two high curvature area bottom and top corner of Zalesak's disk in OPLS method. All the figures are under the resolution of 200.

The interface simulation of the OPLS method is shown in comparison with other methods ICLS, CLS, LS in Zhao [21] (Figure 11). At almost any mesh size, the OPLS method shows an expected result in even every coarse mesh.

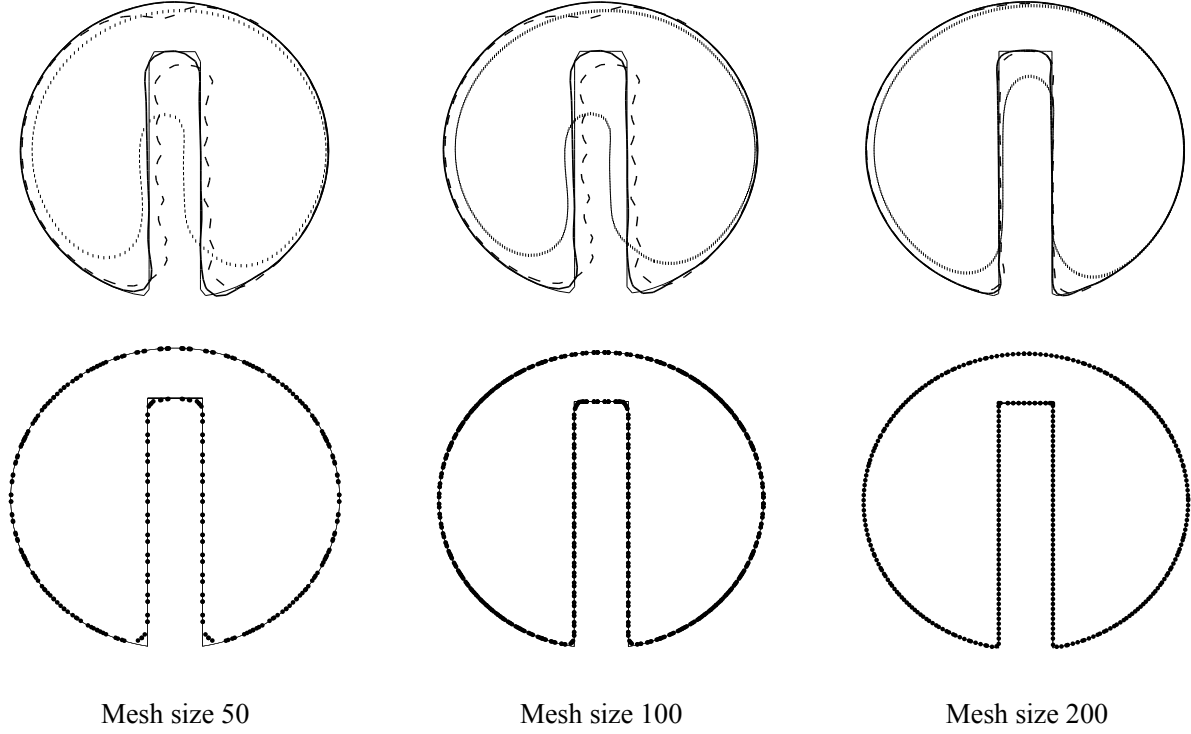


Figure 11. Interface of Zalesak's disk that are described by different methods after two rotations; The light solid line corresponds to the exact solution; Top: ICLS method (dark solid line), CLS method (dashed line), LS method (dotted line); Bottom: OPLS method.

The Zalesak's disk after two completed rotations are shown in Figure 12 at different resolutions (50×50), (100×100) and (200×200). The results are partially enlarged in high curvature regions, such as bottom and top corner of Zalesak's disk. It can be observed that all the results of different meshes coincided with the original shape even every coarse mesh (50×50).

Three locations of $\alpha = (0^\circ, 120^\circ, 240^\circ)$ of rotation angles are shown in Figure 13 to demonstrate the preservation of the disk's shape. The accuracy and robustness of the proposed method are also illustrated by plotting the contour of the level set function location at $\alpha = (0^\circ, 120^\circ, 600^\circ, 720^\circ)$ with different grids mesh: A(50×50), B(100×100) and C(200×200) as shown in Figure 14. As expected, the contour plot of the level set function is

maintained perfectly and only a tiny distortion affects the aforementioned critical region of the corner points, up to the end of the rotation.

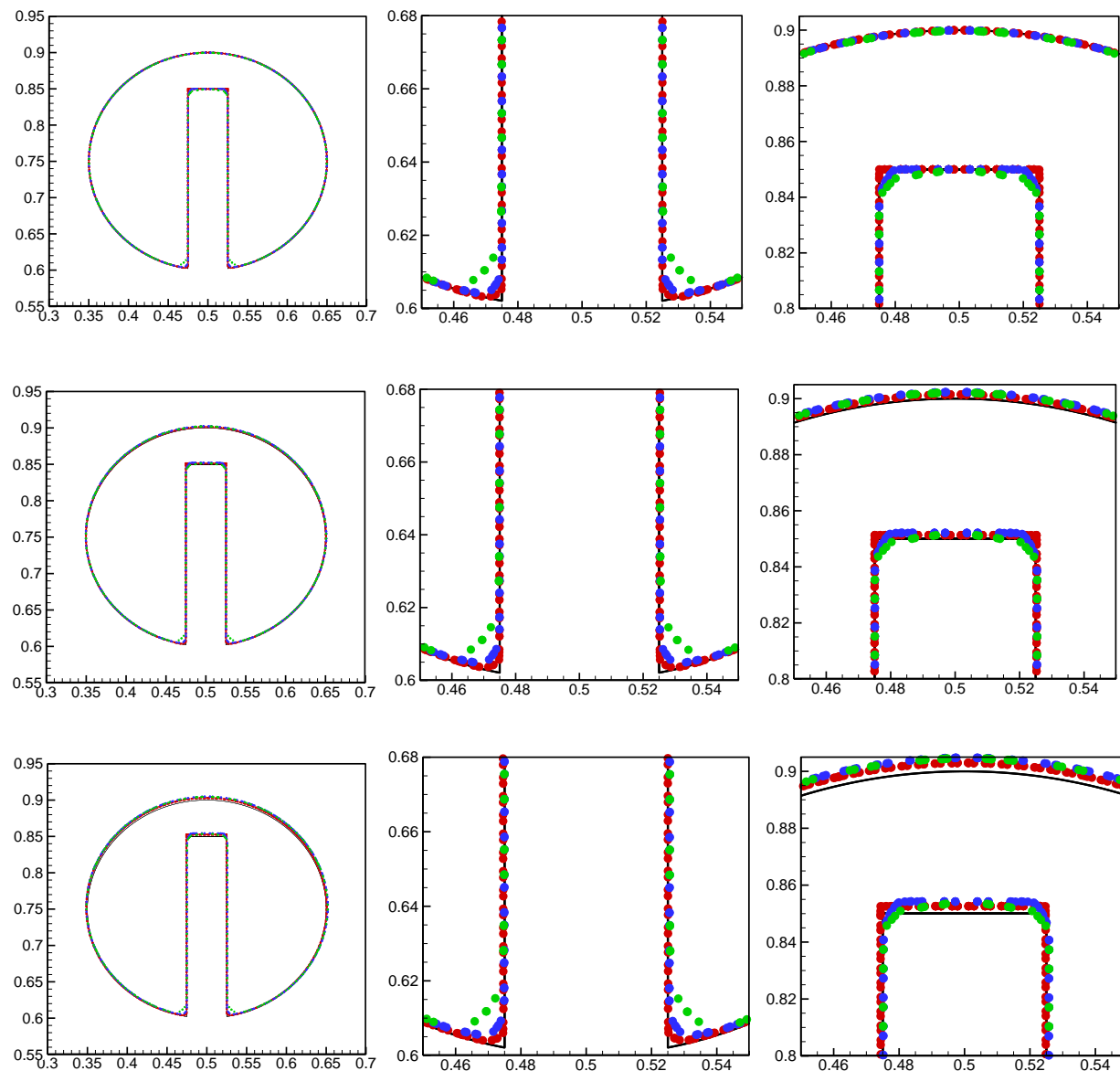


Figure 12. A high quality zoom of Zalesak's disk at bottom and top corner by OPLS method, from top to bottom is beginning $t = 0$; completed one rotation; completed two rotation. In each picture, exact solution (black) is compared with other mesh size: (50×50) (green); (100×100) (blue); (200×200) (red).

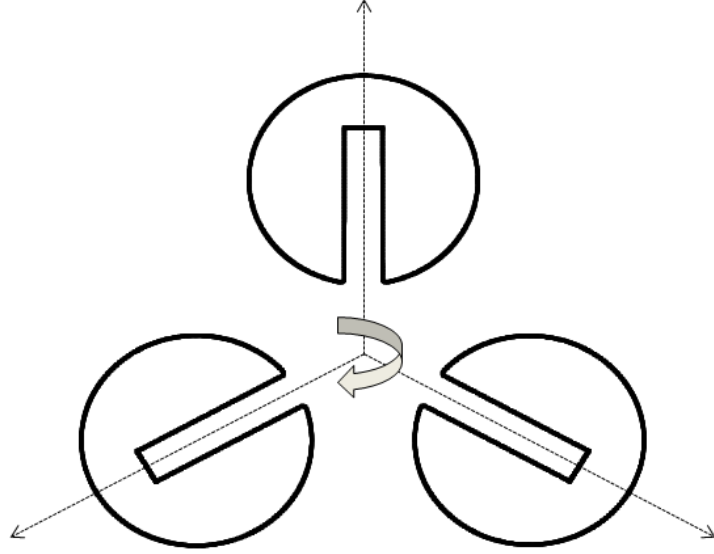


Figure 13. Locations of Zalesak's disk at 0° , 120° , 240° .

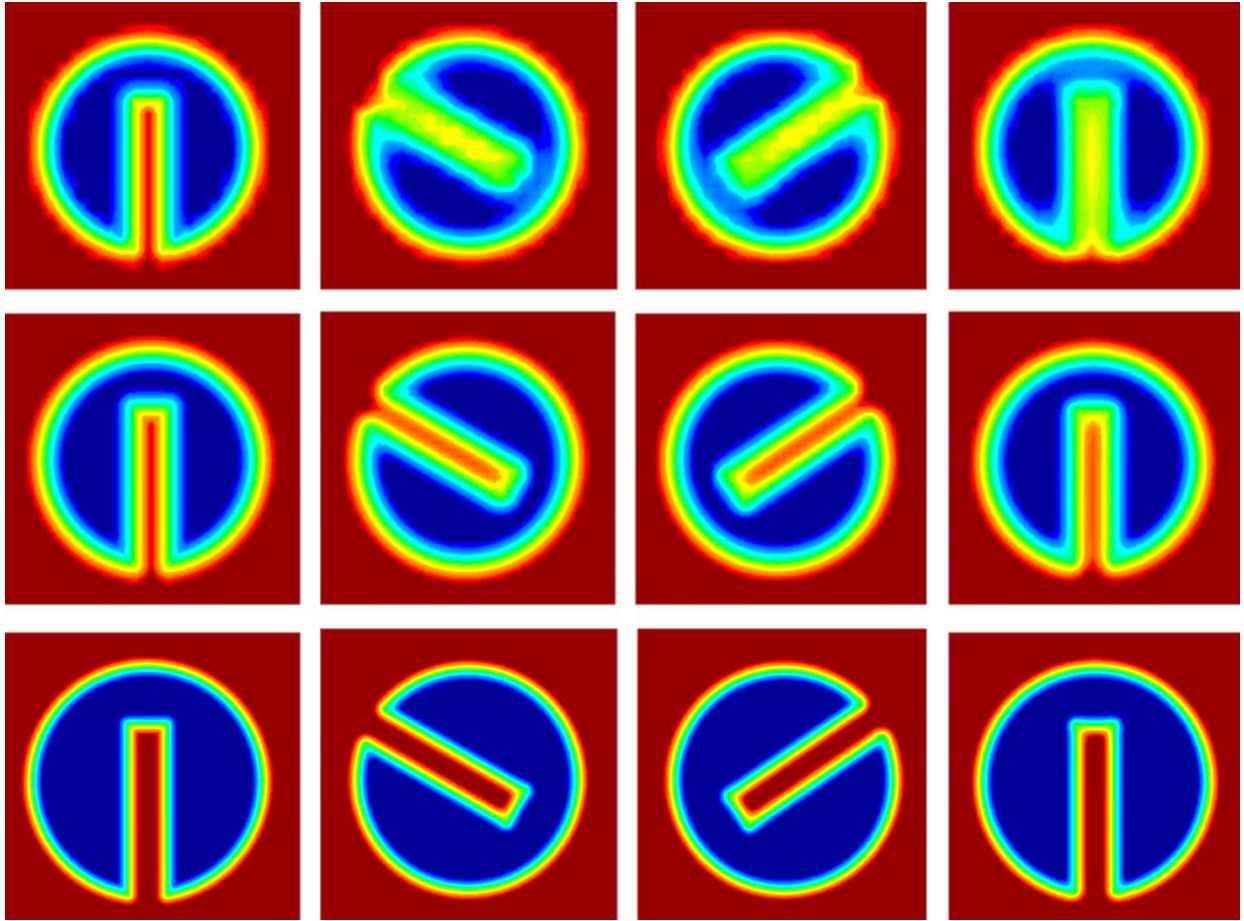


Figure 14. Contour plots of the LS function of Zalesak's disk. Left to right: different angle is selected $\alpha = (0^\circ, 120^\circ; 600^\circ; 720^\circ)$ in holding time completed two rotations; Top to bottom: three types grids mesh A(50×50), B(100×100) and C(200×200).

4.3. Merged of two circles in cross-shape flow

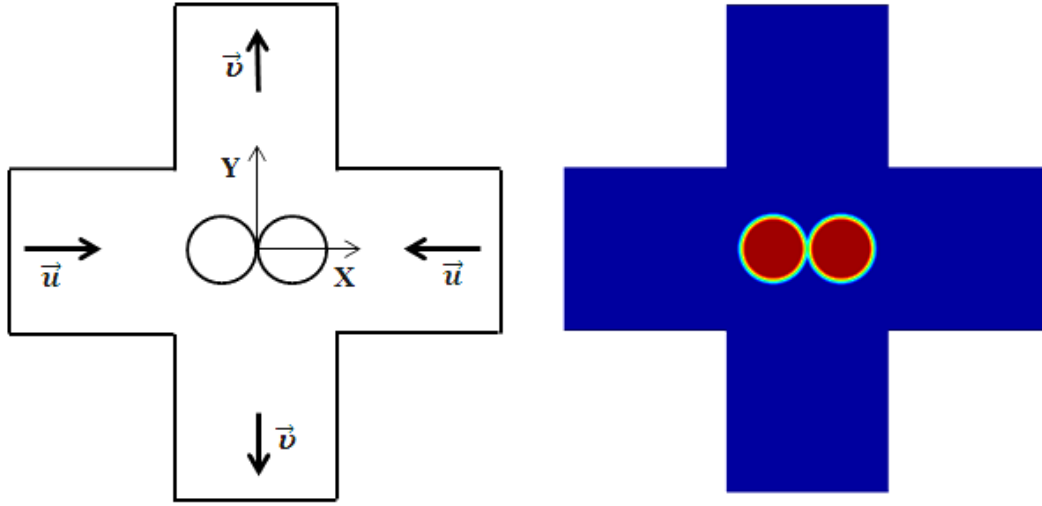


Figure 15. Two circles are merged in cross-shaped flow region and contour level set exhibits at time $t = 0$.

In order to show the ability to manage interface merging by the OPLS method, two circles are tested. The initial two circles of radius $r = 0.2$ centered at $I_1(x, y) = (-0.2, 0)$ and $I_2(x, y) = (0.2, 0)$ are respectively merged into cross-shape flow region with $H = 1$ in Figure 15. The opposite sides of the two circles are carved by two velocities $u = 1$ in direction x and $v = 1$ in direction Y . Snapshot of zero level sets in Figure 16 showed an adequate numerical solution using the OPLS method for managing merging and stretching as time increases. At the position where two circles are merged, the interface disappears and particles are left alone. Therefore each particle that lies at a merged region must be deleted. Then, two circles are stretched to opposite sides. Hence, new particles are generated to account for the stretching. Contour plots for the merging are shown in Figure 17. The process of merging and stretching that are obtained by the OPLS method can be seen in detail in Figure 18. In this test, the deformation reaches maximum stretching without a distortion. It also can be seen in Figure 18 that at final time $t = 30\Delta s$, where $\Delta s = 1/625s$, the relative mass loss is only 0.05%. This test

demonstrates the accuracy and robustness of the OPLS method in managing merging region of interface.

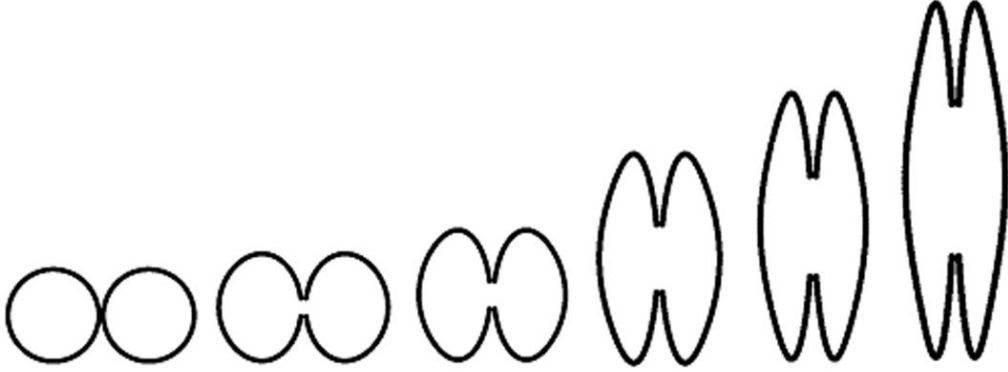


Figure 16. Shape of two circles that are merged at time $t = (0, 5, 10, 20, 25, 30) \times \Delta s$ where $\Delta s = 1 / 625(s)$.

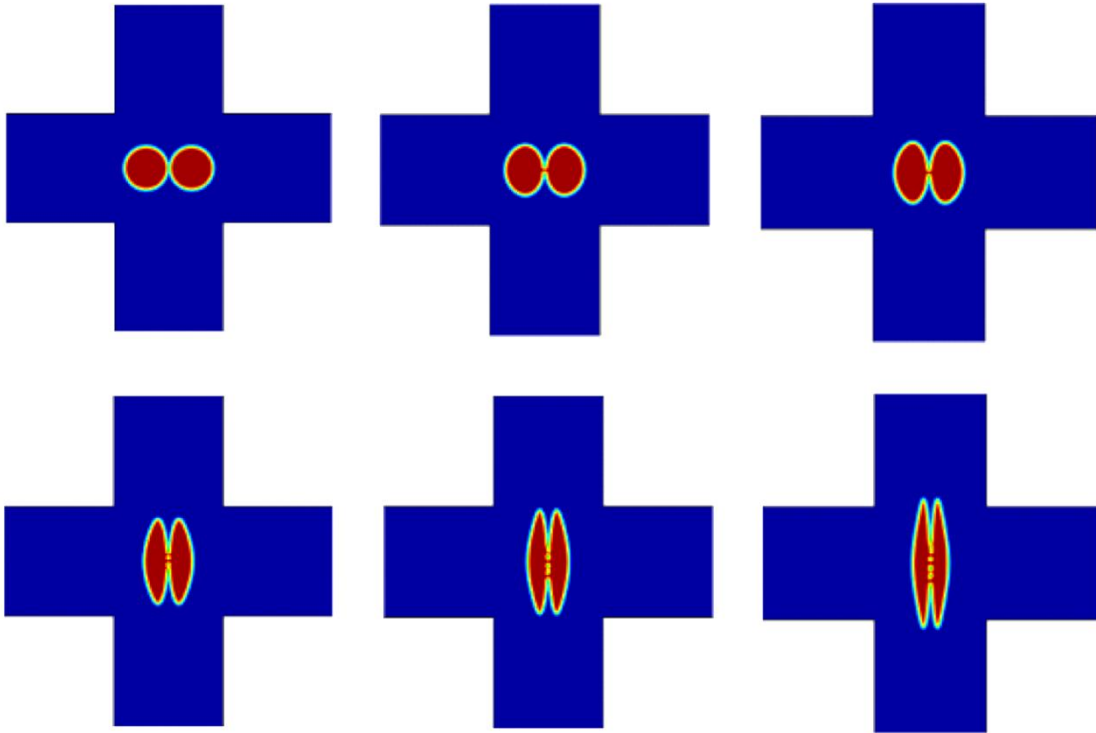


Figure 17. Contour plots of two circles that are merged and stretched in a cross-flow domain over time from left to right, top up bottom $t = (0, 5, 10, 20, 25, 30) \times \Delta s$ where $\Delta s = 1 / 625(s)$.

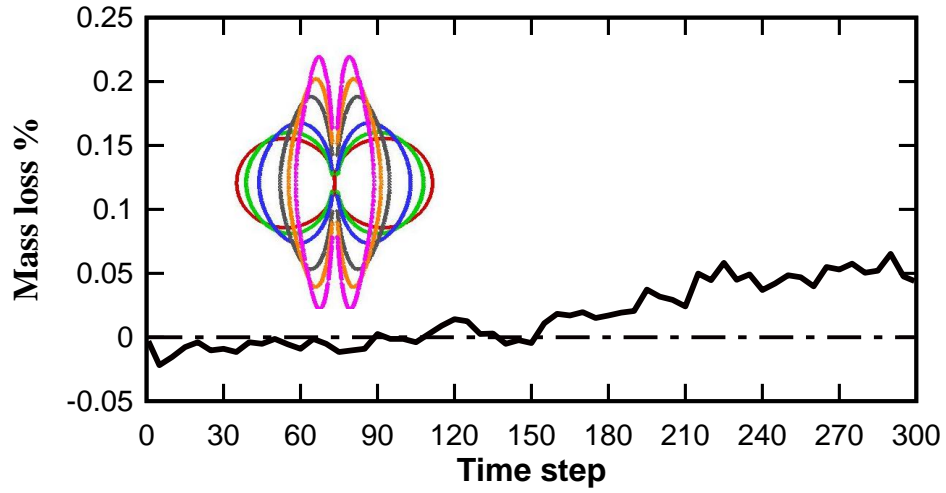


Figure 18. Two circles are merged and stretched at: $t = 0$ (Red); $t = 5$ (Green); $t = 10$ (Blue); $t = 20$ (Gray); $t = 25$ (Orange) and final $t = 30$ (Pink) with unit time $\Delta s = 1/625(s)$ and mass loss respectively.

4.4. Separation of circle

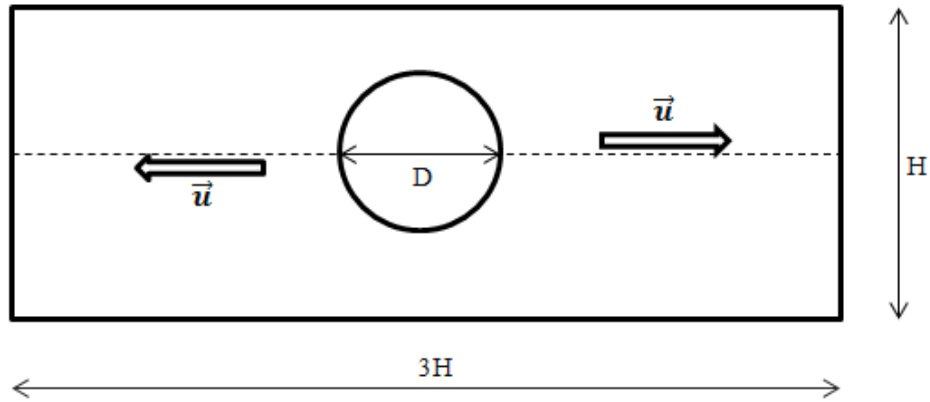


Figure 19. Separation of a circle in a sharp shear flow.

Figure 19 describes a model which is employed to illustrate the ability of OPLS method to manage interface separation by adding particle. The test is given in a rectangular box with $H = 1$. Initially, a circle with diameter $d = 0.6$ is placed at center $I = (0, 0)$ and is separated by the velocity of shear force flow $u = 1$ from two opposite sides of horizontal direction.

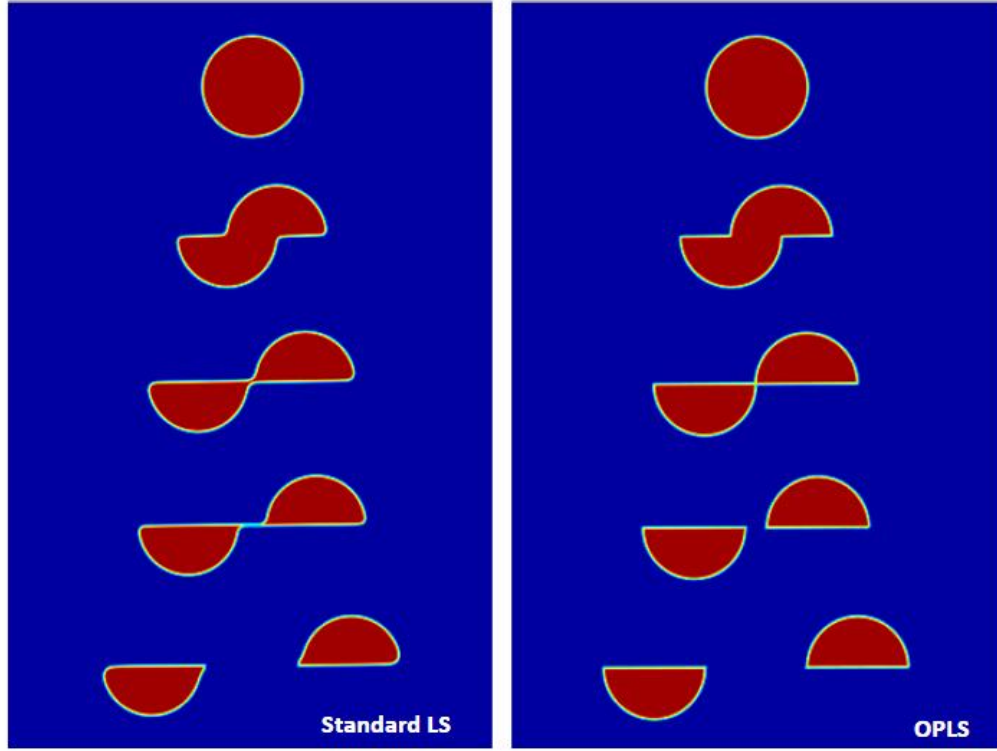


Figure 20. Contour plots of separation of circle at time $t = (0, 40, 85, 100, 150) \times \Delta s$ where $\Delta s = 1/1000(s)$.

The contour plots of Figure 20 exhibits the separation process of original circle into two parts. At each corner, LS standard without re-initialization step shows the incorrect normal vector due to distort in the shape of circle even finer mesh size 256. In comparison, the current OPLS method use a given particle number per cell $N = 2$ in certain ten iterations to support the good results. As expected, no distorted is found in each corner during a long time transform. The ability of particles addition in the OPLS method is shown to accurately describe interfaces undergoing substantial separation. Adding particles during separation process over time is also illustrated by snapshots of circle interface in Figure 21. From the initial time to step 85, when once interface cells are found, particles are generated in each cells and are attracted on the interface. After original circle slipped up into two parts at step 85, no particle is added to final 150 steps.

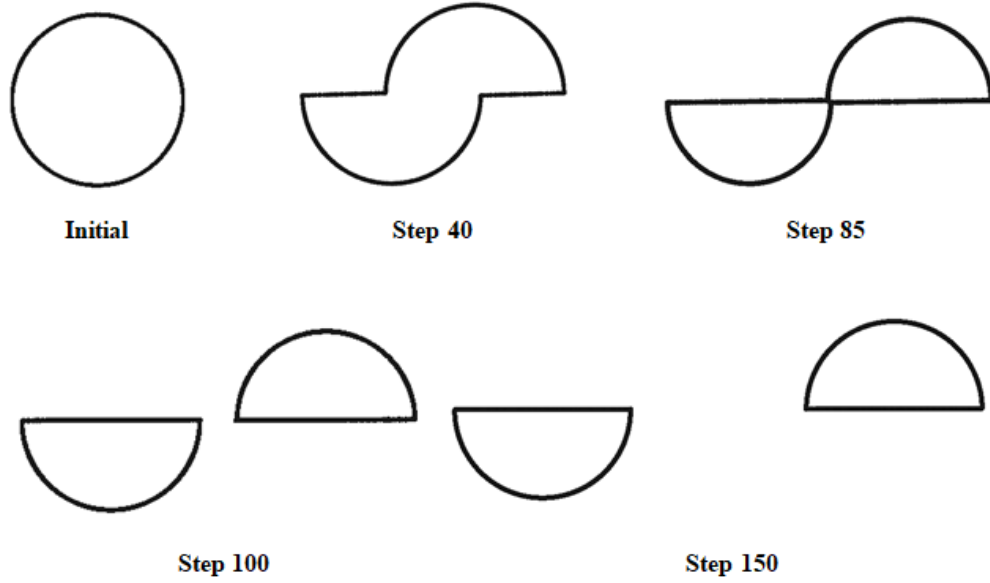


Figure 21. Particles are generated and are attracted in the process of circle separating. After separation produce is completed at step 85 , no particle is added to final 150 steps.

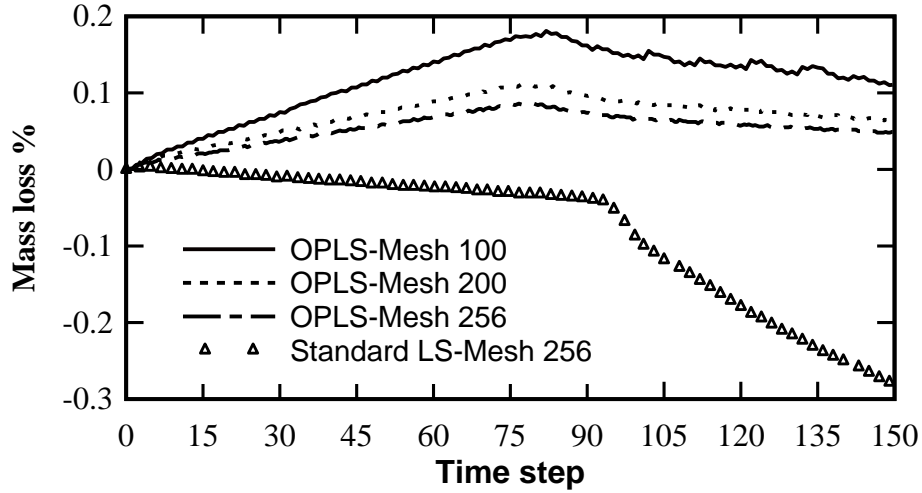


Figure 22. The relative mass loss of OPLS method compares with the standard LS method.

The results of different grid resolutions at different evolution times are shown and compared with standard of LS method in Figure 22. Despite the the complexity and difficulty of this separation problem, the mass loss result from this test is small even in a coarse mesh size. During the first 85 steps, the percentage mass loss result at mesh size 256 of the OPLS method is the best with less than 0.1% . Then it recovers to 0.05% at final step. In contrary, standard of LS

occupy the larger mass loss and the relative mass gain is found to increase to 0.3% at final stage of circle separation even in the finer mesh size. It can be concluded that the accuracy can be improved significantly when solving separating problems with OPLS method.

4.5. Single vortex in a box

The problem ‘‘Vortex-in-a-box’’ exhibits the stretching of a circle which was introduced by Bell in [59]. A circle evolving in a shearing flow is another challenging test for advection scheme of OPLS method to resolve and maintain even thinner filaments. A stream function in a square unit box is given as:

$$\begin{aligned} u(x, y) &= -2 \sin(\pi y) \cos(\pi y) \sin^2(\pi x) \cos(\pi t / T) \\ v(x, y) &= +2 \sin(\pi x) \cos(\pi x) \sin^2(\pi y) \cos(\pi t / T) \end{aligned} \quad (24)$$

Initially a circle with a diameter of 0.3 is placed at $(0.5, 0.75)$. The circle will be transported in the vortex and reach its maximum deformation at $t = 4s$. Then, the velocity components will change their sign and the vortex should reach its initial position and shape at $t = 8s$.

As can be seen in Figure 23, the results from the OPLS method in different grid resolutions at different evolution times are plotted and compared with each other. The interface profile obtained on various mesh sizes (50×50) , (100×100) and (200×200) . At $t = 2s$, the tail section on (50×50) has lost some accuracy due to the mesh size. Once the grid is refined, the accuracy is improved in these grid sizes: (100×100) and (200×200) . When the circle is stretching to maximum deformation, resulted from the coarse case has relatively large deviation from the head, the tail and the interface remains not smooth. However, the result from the smaller resolutions is the better and at the fine mesh (200×200) , the numerical result matches with the exact solution.

Finally, after a long time transform the vortex is then coincided with original shape at $t = 8s$ even every coarse mesh. This is a good result to demonstrate the accuracy and robustness of the OPLS method for such problem.

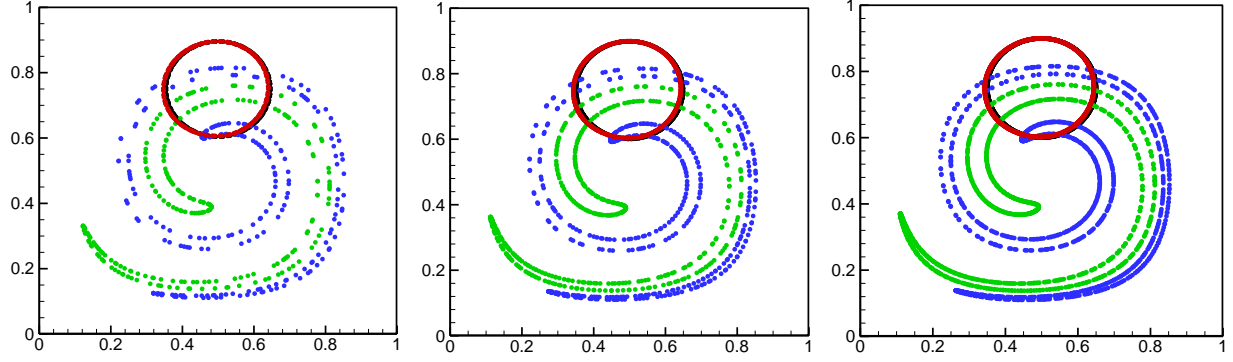


Figure 23. Stretching of circle by a given particle number per cell $N = 2$ in different mesh size: 50×50 ; 100×100 ; 200×200 at $t = 0s$ (black), $t = 2s$ (green), $t = 4s$ (blue) and final $t = 8s$ (red).

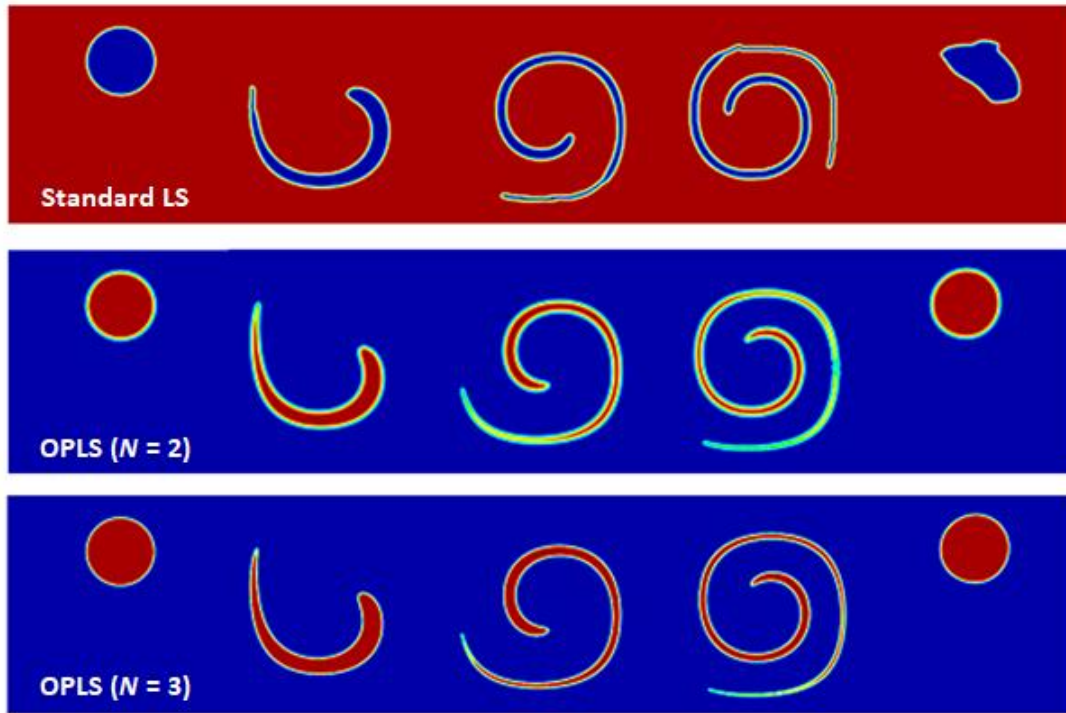


Figure 24. Stretching a circle from left to right at $t = 0s$; $t = 1s$ to the shape at maximum deformation at $t = 4s$ and final return original shape at $t = 8s$ in comparison between Standard LS and OPLS method at mesh size 200×200 .

In order to show the influence of particle number per cell in modification interface, contour plots of stretching circle by standard of the LS method is compared with the OPLS method. In this test, different particle number per cell N is compared and illustrated in Figure 24. It is easy to recognize the weakness of standard LS method in limiting the presentation interface due to the numerical diffusion and the improvements achieved by particles modification interface at high curvature (both nose and tail). For OPLS method, although case $N = 2$ still has relatively tiny deviation at a maximal stretching, the recovered circles are matched nicely to the initial shape at the final process. A thin and elongated filament is well produced on cases $N = 3$ and the shapes at 8s are coincided with the initial circle. It is conceivable that, compared with the standard LS method, the OPLS method exhibits drastically improved interface shape properties.

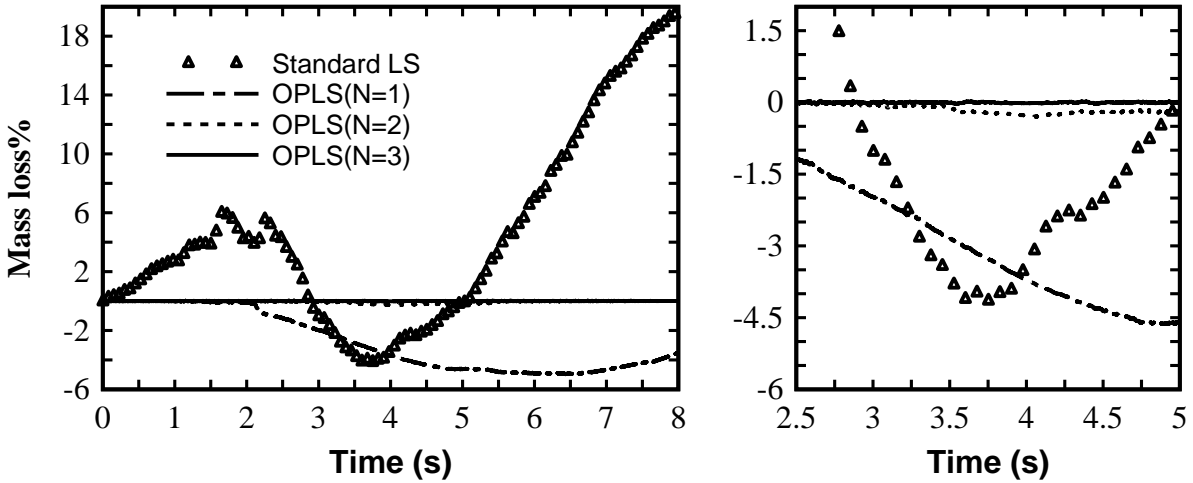


Figure 25. The influence of particle number per cell on mass loss of stretching circle and its zoom.

The effects of the particle number per cell on the conservation mass are shown in Figure 25. Three different cases of number particle per cell $N = 1$; $N = 2$ and $N = 3$ are examined in the same mesh size 200. All the three cases in this study show very good conservation mass after 2s transformation. Then, mass gain peak to 5% at last period of case $N = 1$. The result in cases

$N = 2$ is much better than the former case with 0.31% and very good result has been obtained in case $N = 3$ with 0.03% . A stretching circle in shearing flow is a challenging test to OPLS method. It can be concluded that, the conservation mass problem of standard LS method is corrected successfully by the OPLS method and for such complex problem at least a given number of particle per cell $N = 2$ is suggested.

5. Conclusion

The OPLS method is proposed by combining advantages of the Lagrangian particles and the LS method. In the OPLS method, the LS method is applied to represent the interface smoothly and particles are used to modify the interface directly. The correction procedure based on the Lagrangian particle is applied after calculation of the LS advection and re-initialization steps to address an inherent weakness of the LS method. Considering that only one-layer particles is needed in a given simulation, the OPLS method remarkably reduces the number of particles in comparison with original PLS method, the OPLS method is also able to save computation cost efficiently. Additionally, this method can manage complex topological changes accurately when interfaces merge or stretch.

The OPLS method is applied to several benchmarks cases. The results of the proposed method display an excellent mass conservation property as compared with other existing LS based method. The long term advection of a circle in a uniform field shows that the mass conservation property has been improved significantly in the OPLS compared with the standard LS method. Furthermore, the ability to represent accurate interface and maintain sharpness of corners are demonstrated by the Zalesak's problem. Utilizing different grid resolutions, the OPLS method can give fairly good results, even with the coarse grid. In order to verify the management of merging and separating of interface in under-resolved regions, two tests of circles were set,

including cross-shape flow and separation. Despite the complexity and difficulty, the results from those tests were good. In the example of vortex in a box, while standard LS methods produced a distorted interface due to the inaccuracy, the OPLS shows an influence of particles correction on curvature estimation without small droplets separating from the main structure. It has been shown that the proposed method could capture the small scale structures very well.

According to the merits of the OPLS method mentioned above, it can be used to capture bubbles or droplets with a moderate number of grid cells without mass loss and shape distortion. The proposed method can be further extended to multiphase flows in a straightforward way and future work will essentially focus on the application of the method to three-phase flow problems.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 51279050), the Fundamental Research Funds for the Central Universities (No. 2016B05014), Fok Ying-Tong Education Foundation for Young Teachers in the Higher Education Institutions of China (No. 151073), Ministry of Water Resources non-profit specific industry appropriation (No. 201501036, No. 201501034 and No. 201501033), Qing Lan Project and China Scholarship Council.

Reference

1. Belytschko T, Guo Y, Liu WK, Xiao SP, *A unified stability analysis of meshless particle methods*. Int J Numer Meth Eng, (2000). **48**: 1359–400.
2. Marrone S, Colagrossi A, Le Touze D, Graziani G, *Fast free-surface detection and level-set function definition in SPH solvers*. J Comput Phys, (2010). **229**: 3652–63.
3. Bockmann A, Shipilova O, Skeie G, *Incompressible SPH for free surface flows*. Comput Fluids, (2012). **67**: 138–51.
4. Monaghan, JJ, *Smoothed particle hydrodynamics*. Reports on Progress in Physics, (2008). **68**: 1703–1759.

5. Xenakis AM, Lind SJ, Stansby PK, Rogers BD, *An incompressible SPH scheme with improved pressure predictions for free-surface generalised Newtonian flows*. J Non-Newtonian Fluid Mech, (2015). **218**: 1-15.
6. Khorasanizade SH, Sousa JMM, *An adaptive fully-Lagrangian meshless method for incompressible laminar flow airfoil studies*. Aerospace Science and Tech, (2017). **64**: 161-170.
7. Unverdi SO, Tryggvason G, *A front-tracking method for viscous, incompressible, multifluid flows*. J Comput Phys, (1992). **100**: 25-37.
8. Ciortan C, Wanderley JBV, GuedesSoares C, *Free surface flow around a ship model using an interface-capturing method*. Ocean Eng, (2012). **44**: 57-67.
9. Welch JE, Harlow FH, Shannon JP, Daly BJ, *The MAC method*. Technical Report LA-3425, Los Alamos National Laboratory, (1965).
10. Harlow F, Welch JE, *Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface*. Comput & Fluids, (1965). **37**: 907-930.
11. McKee S, Tome MF, Ferreira VG, Cuminato JA, Castelo A, Sousa FS, Mangiavacchi N, *Review the MAC method*. Comput & Fluids, (2008). **37**: 907-930.
12. Amsden A, Harlow F, *The SMAC method: a numerical technique for calculating incompressible fluid flows*. Technical Report LA-4370, Los Alamos National Laboratory, (1970).
13. Deville, MO, *Numerical experiments on the MAC code for slow flow*. J Comput Phys (1974). **15**: 362-74.
14. Sousa FS, Mangiavacchi N, Nonato LG, Castelo A, Tome MF, McKee S, *A front-tracking/front-capturing method for the simulation of 3D multi-fluid flows with free surfaces*. J Comput Phys, (2004). **198**: 469-499.
15. Hirt C, Nichols B, *Volume of fluid (VOF) method for the dynamics of free boundaries*. J Comput Phys, (1981). **39**: 201-225.
16. Pilliod JE, Puckett EG, *Second-order accurate volume-of-fluid algorithms for tracking material interfaces*. J Comput Phys, (2004). **199**: 465-502.
17. Osher S, Sethian JA. , *Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations*. J Comput Phys, (1988). **79**: 12-49.
18. JA, Sethian, *Level set methods and fast marching methods*. Cambridge: Cambridge university press, (1999).
19. Herreros M.I, Mabssout, M, Pastor M, , *Application of level-set approach to moving interfaces and free surface problems in flow through porous media*. Comput Methods Appl Mech Eng, (2006). **195**: 1-25.
20. Quecedo M, Pastor M, *Application of the level set method to the finite element solution of two-phase flows*. Int J For Numer Meth in Eng, (2001). **30**: 645-664.
21. Zhao LH, Mao J, Bai X, Liu XQ, Li TC, Williams JJR, *Finite element implementation of an improved conservative level set method for two-phase flow*. Comput Fluids (2014). **100**: 138-154.
22. Losasso F, Fedkiw R, Osher S, *Spatially adaptive techniques for level set methods and incompressible flow*. Comput & Fluids, (2006). **35**: 995-1010.
23. Jiang Liang, Liu Fengbin, Chen Darong, *A fast particle level set method with optimized particle correction procedure for interface capturing*. J Comput Phys, (2015). **299**: 804-819.
24. Sussman, *An efficient interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow*. SIAM J Sci Comput, (1999). **20**: 1165-1191.
25. Peng D, Merriman B, Osher S, Zhao H, Kang M, *A PDE-based fast local level-set method*. J Comput Phys, (1999). **155**: 410-438.
26. Russo G, Smereka P, *A remark on computing distance functions*. J Comput Phys, (2003). **163**: 51-67.

27. Elin Olsson, Gunilla Kreiss, *A conservative level set method for two phase flow*. J Comput Phys, (2005). **210**: 225–246.
28. Elin Olsson, Gunilla Kreiss, Sara Zahedi, *A conservative level set method for two phase flow II*. J Comput Phys, (2007). **225**: 785–807.
29. Sheu Tony WH. , Yu CH, Chiu PH, *Development of level set method with good area preservation to predict interface in two-phase flows*. Int J Numer Meth Fluids, (2011). **67**: 109–134.
30. Desjardins O, Moureau V, Pitsch H, *An accurate conservative level set/ghost fluid method for simulating turbulent atomization*. J Comput Phys, (2008). **227**: 8395–8416.
31. Barto PT, Obadia B, Drikakis D, *A conservative level-set based method for compressible solid/fluid problems on fixed grids*. J Comput Phys, (2011). **230**: 7867–7890.
32. Owkes M, Desjardins O, *A discontinuous Galerkin conservative level set scheme for interface capturing in multiphase flows*. J of Comput Phys, (2013).
33. Sara Zahedi, Katarina Gustavsson, Gunilla Kreiss, *conservative level set method for contact line dynamics*. J Comput Phys, (2009). **228**: 6361–6375.
34. Kees CE, Akkerman I, Farthing MW, Bazilevs Y, *A conservative level set method suitable for variable-order approximations and unstructured meshes*. J Comput Phys, (2011). **230**: 4536–4558.
35. Sato Y, Niceno B, *A new contact line treatment for a conservative level set method*. J Comput Phys, (2012). **231**: 3887–3895.
36. van der Pijl SP, Segal A, Vuik C, Wesseling P, *A mass-conserving Level-Set method for modelling of multi-phase flows*. Int J For Numer Meth Fluids (2005). **47**: 339–361.
37. Sussman M, Puckett EG, *A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows*. J Comput Phys, (2000). **162**: 301–337.
38. Enright D, Fedkiw R, Ferziger J, Mitchell L, *A hybrid particle level set method for improved interface capturing*. J Comput Phys, (2002). **183**: 83–116.
39. Wang Z, Yang J, Stern F, *An improved particle correction procedure for the particle level set method*. J Comput Phys, (2009). **228**: 5819–5837.
40. Hieber SE, Koumoutsakos P, *A Lagrangian particle level set method*. J Comput Phys, (2005). **162**: 342–367.
41. Gaudlitz D, Adams NA, *On improving mass-conservation properties of the hybrid particle-level-set method*. Comput & Fluids, (2008). **37**: 1320–1331.
42. Lanniello S, Di Mascio A, *A self-adaptive oriented particles Level-Set method for tracking interfaces*. J Comput Physics, (2009). **229**: 1353–1380.
43. Chen Q, Wang J, Liu K, *Improved CE/SE scheme with particle level set method for numerical simulation of spall fracture due to high-velocity impact*. J Comput Phys, (2010). **229**: 7503–7519.
44. Pang S, Chen L, Zhang M, Yin Y, Chen T, Zhou J, Liao D, *Numerical simulation two phase flows of casting filling process using SOLA particle level set method*. App Math Model, (2010). **34**: 4106–4122.
45. Magnus Vartdal, Arne Bøckmann, *An oriented particle level set method based on surface coordinates*. J Comput Phys, (2013). **251**: 237–250.
46. Philip J. Archer, Wei Bai, *A new non-overlapping concept to improve the Hybrid Particle Level Set method in multi-phase fluid flows*. J Comput Phys, (2015). **282**: 317–333.
47. Kamran K, Rossi R, Onate E, *A locally extended finite element method for the simulation of multi-fluid flows using the Particle Level Set method* Comput Meth Appl Mech Eng, (2015). **294**: 1-18.
48. Juan, Luis Prieto, *A multiscale, particle level set method for Newtonian and non-Newtonian interface flows*. Comput Meth Appl Mech Eng, (2016). **307**: 164–192.
49. Wang Z, Yang J, Stern F, *An improved particle correction procedure for the particle level set method*. J Comput Phys, (2009). **228**: 5819–5837.

50. Zhaorui Li, Farhad A. Jaber and Tom I-P. Shih, *A hybrid Lagrangian–Eulerian particle-level set method for numerical simulations of two-fluid turbulent flows*. Int J For Numer Meth Fluids, (2008). **56**: 2271-2300.
51. Juric, Seungwon Shin and Damir, *A hybrid interface method for three-dimensional multiphase flows based on front tracking and level set techniques*. Int J For Numer Meth Fluids, (2009). **60**: 753-778.
52. Wang Y, Simakhina S, Sussman M, *A hybrid level set-volume constraint method for incompressible two-phase flow*. J Comput Phys, (2012). **231**: 6438-6471.
53. Taylor RL, Zhu JZ, *The Finite Element Method: Its Basis and Fundamentals*. (1967).
54. Shu C, Osher S, *Efficient implementation of essentially nonoscillatory shock capturing schemes*. J Comput Phys, (1998). **77**: 439.
55. Kang M, Fedkiw R, and Liu XD, *A boundary condition capturing method for multiphase incompressible flow*. J Sci Comput, (2000). **15**: 323.
56. J, Peraire, *A Finite Element Method for Convection Dominated Flows*. PhD thesis, University of Wales, Swansea, (1986).
57. Zienkiewicz OC, Taylor RL, Nithiarasu P, *The Finite element method for fluid dynamics, Elsevier, Sixth edition*. (2005).
58. Zalesak, ST, *Fully multidimensional flux-corrected transport algorithms for fluids*. J Comput Phys (1979). **31**: 335–362.
59. Bell J, Colella P, Glaz H,, *A second-order projection method for the incompressible Navier–Stokes equations*. JComput Phys, (1989). **85**: 257–83.